Identifying Predictive Structures in Relational Data Using Multiple Instance Learning

Amy McGovern AMY@CS.UMASS.EDU
David Jensen JENSEN@CS.UMASS.EDU
Knowledge Discovery Laboratory, Computer Science Dept., Univ. of Massachusetts Amherst, Amherst, MA 01003, USA

Abstract

This paper introduces an approach for identifying predictive structures in relational data using the multiple-instance framework. By a predictive structure, we mean a structure that can explain a given labeling of the data and can predict labels of unseen data. Multiple-instance learning has previously only been applied to flat, or propositional, data and we present a modification to the framework that allows multiple-instance techniques to be used on relational data. We present experimental results using a relational modification of the diverse density method (Maron, 1998; Maron & Lozano-Pérez, 1998) and of a method based on the chi-squared statistic (McGovern & Jensen, 2003). We demonstrate that multiple-instance learning can be used to identify predictive structures on both a small illustrative data set and the Internet Movie Database. We compare the classification results to a *k*-nearest neighbor approach.

1. Introduction

Identifying useful structures in large relational databases is a difficult task. For example, consider the task of predicting which movies will be nominated for academy awards every year. The Internet Movie Database (IMDb) contains about one hundred movies that were nominated for academy awards in the time period 1970 to 2000 and thousands of movies that were not nominated in this time period. We would like to identify relational structure from a set of positive and negative examples (e.g., the structure surrounding nominated and non-nominated movies) that can explain known labels and predict labels for unseen data. Specifically, given the schema for the IMDb shown in Figure 1, we would like to identify some substructure that can predict which movies will be nominated and which movies will not be nominated. An example substructure could be a movie where one of the actors was previously nominated for an academy award. Such structures are useful not only for classification and prediction tasks but also for better understanding of large relational databases.

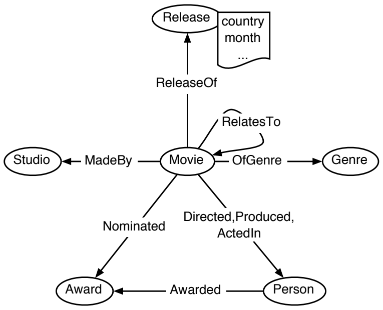


Figure 1. Schema that we used for the IMDb

Multiple instance learning (MIL) (details of MIL are given in Section 2) is a promising framework for identifying predictive structures in large relational databases. First, MIL methods are designed for learning from ambiguous and partially labeled data. With relational data, it is often easy to label a collection of objects and their relations. However, labeling each individual object and relation by its contribution to the overall situation is more difficult. For example, we can obtain the labels for the movie subgraphs by noting whether the movie was nominated for an academy award, but it would be difficult to label each actor and studio by their individual contribution to whether the movie was nominated for an award. Second, multiple-instance (MI) techniques are designed to identify which part of the data can explain the labels. For example, the relations in the movies example could contain all related movies, releases, studios, etc., for each nominated movie, but the best concept might only use the studio and producers linked via a movie.

MIL has been used successfully in a number of applications using propositional data (Amar et al., 2001; Dietterich et al., 1997; Goldman et al., 2002; Maron, 1998; Maron & Ratan, 1998; Zhang & Goldman, 2002; Zucker

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2003**	2. REPORT TYPE	3. DATES COVERED **00-00-2003 to 00-00-2003**

4. TITLE AND SUBTITLE **Identifying Predictive Structures in Relational Data Using Multiple Instance Learning**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Massachusetts Amherst,Knowledge Discovery Laboratory,140 Governors Drive,Amherst,MA,01003**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**8**	

& Chevaleyre, 2000). However, none of these techniques have examined MIL approaches for relational data even though the data set used in the introductory MIL paper (Dietterich et al., 1997) was relational (it was flattened into feature vectors to solve the task). By working with the data in relational form, we can detect structures that cannot be represented in a feature vector format. For example, a link between two related movies where the movie's producer was also nominated for an academy award for a previous movie would be very difficult to represent in propositional data, especially if the form of the final structure is not known in advance. Simply flattening the relational data (into homogeneous feature vectors) presents a number of problems. The homogeneity of the resulting data either means data duplication (which will affect probability estimation) or data loss through aggregation.

2. Notation and background

We use the PROXIMITY[1] system to represent, store, and query relational data sets. Let $G = (v, e)$ be a graph. Objects in the world, such as people, places, things, and events, are represented as vertices in the graph. Relations between these objects, such as *acted-in(actor, movie)* are represented by edges. In general, if there is a relation $r(o_1, o_2)$, then $o_1, o_2 \in v$ and $r \in e$. In PROXIMITY, vertices are called objects and relations are called links. Both objects and links can have multiple attributes associated with them. For example, using the schema shown in Figure 1, movies, people, genres, etc., are all objects. Relationships such as *awarded(movie, best-picture)* are links. Attributes can be associated with objects, such as *movie.name*, or links, such as *awarded.award-status*. PROXIMITY allows us to query the database using a graphical query language called qGraph (Blau et al., 2002). qGraph provides a form of abstraction on top of SQL by allowing us to construct visual queries of the graphical database. Queries return a collection of subgraphs and not just a set of database rows.

An MI learner uses labeled *bags* where a bag is a collection of *instances* with one label for the entire collection. A *positive bag* contains at least one instance of the *target concept* while a *negative bag* contains none. With flat data, both instances and target concepts are points in feature space. With relational data, both instances and target concepts are graphs, or relations among a (heterogeneous) set of objects. The goal is to find a concept that explains the labels for the bags and can be used to predict labels for unseen data. It is not known in advance which instance caused the bag to be labeled as positive. If this were known, a supervised learning approach could be used instead.

We present an approach to adapting the MIL framework for use with relational data where bags are collections of graphs. The instances in the bag can be either explicitly enumerated as a set of graphs or they can be the set of (implicit) subgraphs of a single, larger, graph. The structure of the relational data determines which representation is most appropriate. If the data consist of sets of disjoint graphs, such as the MUSK task where each conformation of a molecule could be represented as a separate graph, then it is better to explicitly enumerate the instances in each bag. If the data consist of a large connected database, such as IMDb, then a bag consisting of single large graph can be easily created by querying the database. For example, in IMDb, the bags for the movies nominated for academy awards can be created by querying for all objects connected to a nominated movie by one or two links.

For the MI notation, we follow that of Maron (1998) and Maron and Lozano-Pérez (1998). The set of positive bags is denoted B^+ and the ith positive bag is B_i^+. Likewise, the set of negative bags is denoted B^- and the ith negative bag is B_i^-. If the discussion applies to both types of bags, we drop the superscript and refer to it as B. The jth instance of the ith bag is denoted B_{ij}. The target concept is denoted c_t and other concepts as c. With flat data, a concept is a point in feature space. With relational data, a concept is an attributed graph.

3. MI learning on relational data

The data available to an MI learner is a set of positive and negative bags, B^+ and B^-. If the concept is a feature vector v, then each bag consists of a set of feature vectors: $B_i = \{v_{i1}, v_{i2}, \ldots, v_{ik}\}$. The most straightforward transformation to apply MIL to relational data is to have each instance represented as a separate graph. In this case, a bag would consist of a set of graphs: $B_i = \{G_{i1}, G_{i2}, \ldots, G_{ik}\}$. The goal is then to find a concept that can explain the labeling of the bags. The concept, c, is a subgraph of one the graphs in B^+ and B^-. This representation is best suited for tasks where the data are already available as a set of disjoint graphs. The MUSK data set (Dietterich et al., 1997), image recognition tasks (Maron & Ratan, 1998), and the mutagenesis data set (Zucker & Chevaleyre, 2000) fit into this framework.

When the relational data are available as a large connected graph instead of a set of unconnected graphs, it may be easier to identify a single subgraph as containing something positive instead of enumerating every instance. For example, in the IMDb, we can hypothesize that there is some relational structure surrounding movies that could be used to predict whether a movie gets nominated for an academy award. Without knowing the structure in advance, it would be very difficult to create bags of every possible struc-

[1] For additional details on PROXIMITY, see http://kdl.cs.umass.edu.

ture. However, it is relatively easy to identify the depth-two structure surrounding the movies and to use this to create bags where each bag has only one graph. The instances are assumed to be the set of all subgraphs of the single graph in the bag.

More formally, we propose to create the set of bags B^+ and B^- such that $B_i = \{G_i\}$ where G_i is a single (large) graph. The instances of B_i are assumed to be the set of all subgraphs of G_i. Since the size of this set is exponential in the size of G_i, where $|G_i|$ is defined as the sum of the number of vertices and edges in G_i, we do not explicitly enumerate the instances for each bag. Instead, the search methods take into account this assumption.

3.1. Relational diverse density

Several existing MI methods can be transformed to work with relational data. We adapt both diverse density (Maron, 1998; Maron & Lozano-Pérez, 1998) and chi-squared (McGovern & Jensen, 2003). We first briefly review the definitions for diverse density. The most diversely dense concept is defined as that which is closest to the intersection of the positive bags and farthest from the union of the negative bags. More precisely, Maron defines the diverse density of a particular concept c to be: $DD(c) = P(c = c_t|B^+, B^-)$. We refer to $P(c = c_t)$ as $P(c)$ to simplify the equations. Using Bayes rule and assuming independence, this can be reduced to finding the concept c for which the likelihood: $\prod_{1 \leq i \leq n} P(c|B_i^+) \prod_{1 \leq i \leq m} P(c|B_i^-)$ is maximal. The probability that concept c is the target concept given the evidence available in the bag, $P(c|B_i)$, still needs to be determined. Maron discusses several ways to do this. In this work, we follow his suggestion of using a noisy-or model (Pearl, 1988), in which case we have:

$$P(c|B_i^+) = 1 - \prod_{1 \leq j \leq p} (1 - P(B_{ij}^+ \in c)) \qquad (1)$$

$$P(c|B_i^-) = \prod_{1 \leq j \leq p} (1 - P(B_{ij}^- \in c)), \qquad (2)$$

where p is the number of instances in bag B_i and $P(B_{ij} \in c)$ is the probability that the specified instance is in the concept.

Calculating $P(B_{ij} \in c)$ requires a specific form of target concept. In the case of flat data, Maron often used what he called the single-point concept which is a point in feature space. With this concept, the calculation of $P(B_{ij} \in c)$ is based on the Euclidean distance between points B_{ij} and c in feature space. We need to define $P(B_{ij} \in c)$ when B_{ij} and c are both attributed graphs instead of points in feature space. To do this, we need a method for measuring the distance between two attributed graphs.

Metrics for measuring the distance between attributed graphs are not as well studied as metrics for flat data. We use the metric proposed by Bunke and Shearer (1998) which is based on finding the maximal common subgraph (MCS) between two graphs. They demonstrate that this distance measure satisfies the metric properties. The distance between two graphs G_1 and G_2 is defined as:

$$d(G_1, G_2) = 1 - \frac{|MCS(G_1, G_2)|}{\max(|G_1|, |G_2|)} \qquad (3)$$

where $MCS(G_1, G_2)$ is the maximum common subgraph of G_1 and G_2. This metric was developed for unlabeled graphs but can be modified so that the MCS also uses the attributes to limit the number of matches. A disadvantage of this metric is that computing the MCS is exponentially complete. In the course of a thorough search in concept space, MCS is calculated frequently. We approximate the calculation by limiting the depth of the recursive search. Research on a principled polynomial-time distance metric for attributed graphs is a topic for future work. Based on this metric, we define $P(B_{ij} \in c)$ as:

$$P(B_{ij} \in c) = \frac{|MCS(B_{ij}, c)|}{\min(|B_{ij}|, |c|)} \qquad (4)$$

Note that Equation 4 is a slight modification of Equation 3 where the maximum is replaced by a minimum. Since we are searching for the best subgraph, it is better to weight the match by the size of the proposed subgraph rather than by the size of the instances or of the bag, which could be arbitrarily large. If the instances in the bag are not enumerated, $P(c|B_i)$ becomes:

$$P(c|B_i^+) = \frac{|MCS(c, B_i^+)|}{\min(|c|, |B_i^+|)} \qquad (5)$$

$$P(c|B_i^-) = 1 - \frac{|MCS(c, B_i^-)|}{\min(|c|, |B_i^-|)}. \qquad (6)$$

This means that the probability that c is the correct concept given the evidence available in positive bag B_i^+ is the percent match of graph c to graph B_i^+. Likewise, the probability that c is the correct concept given the evidence in negative bag B_i^- is one minus the percent match of graph c to graph B_i^-. In other words, if c matches highly with B_i^+, the probability that c is correct will be high but if it matches highly with B_i^-, the probability that c is correct will be low.

3.2. Relational chi-squared method

In addition to diverse density, we present results using the chi-squared MI method (McGovern & Jensen, 2003). Chi-squared is simpler to calculate than diverse density and it allows for a more thorough search of the concept space because it provides a guaranteed pruning method. Chi-squared is calculated by filling in the contingency table shown in Table 1. The rows of the table correspond to the

Table 1. Contingency table used by the chi-squared method. The cells are filled in using the predicted and known labels for the training bags using the proposed concept.

		Actual Bag label	
		+	-
Predicted bag label	+	a	b
	-	c	d

predicted label from the concept and the columns correspond to the actual labels for the training bags. Assuming a method for labeling the bags given a proposed target concept, the table is filled out in the following manner. If the concept predicts that the bag will be positive and it is positive, *a* is incremented. If the prediction is positive but the bag is really negative, *b* is incremented. If the prediction is negative and the bag is positive, *c* is incremented. Finally, *d* is incremented if the concept predicts negative and the bag is negative. Chi-squared is calculated by summing the squared differences for the expected values in each cell of the contingency table versus the observed values.

The best concept is defined as that with the highest chi-squared value. This will occur when the mass is concentrated on the main diagonal (e.g., in *a* and *d*) which means that the concept is predicting the most positive and the most negative bags correctly. More information about the chi-squared evaluation function for MIL can be found in (McGovern & Jensen, 2003).

4. Experimental results: illustrative data set

We first present results using a small illustrative database where we both know the target answer in advance and can easily visualize the data. The objects and links each have one real-valued attribute associated with them. The target concept, shown in Figure 2, is a size-three clique with a particular set of attribute values on the objects and links.

We illustrate both chi-squared and diverse density using both data representations and this target concept. In both cases, graphs, including objects, links, direction of the links, and attributes, were generated randomly. To create a positive instance, a graph was randomly grown from the target clique. Negative instances were randomly grown from an empty graph. Attribute values from the target concept can be used in negative instances so long as the entire concept is not included. For the first data representation, both positive and negative graphs varied in size from three to ten objects with the same number of random links. Each positive bag had one positive instance and from two to six negative instances. Negative bags contained from three to seven negative instances. A sample positive instance and a sample negative instance are shown in Figure 3a. The bags for the second data representation, which contained only one instance per bag, varied in size from ten to twenty objects and had twice as many random links as there were objects. Example positive and negative bags for this framework are shown in Figure 3b. In both cases, we generated twenty positive bags and twenty negative bags.

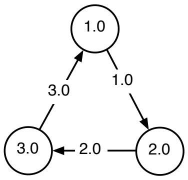


Figure 2. Target concept for the illustrative data set

For this experiment, we compared the relative prediction accuracies for the relational diverse density approach, the chi-squared technique, and the *k*-nearest neighbors (kNN) method. We repeat this comparison for both data representations. For the diverse density and chi-squared approaches, the MI learner identified the best concept (or set of concepts) for predicting the bag labels. Given a relational concept c and a bag B_i with an unknown label, the predicted real-valued label is:

$$\text{label} = \max_{1\leq j\leq k} P(B_{ij} \in c) \quad = \quad \max_{1\leq j\leq k} \frac{|MCS(B_{ij},c)|}{\min(|B_{ij}|,|c|)}.$$

If there is only one graph in the bag, this becomes:

$$\text{label} = \frac{|MCS(B_i,c)|}{\min(|B_i|,|c|)}.$$

Under this formulation, the predicted label for the bag will be a real number in the interval $[0,1]$. A prediction of zero means the bag should be labeled as negative and a prediction of one means that the bag should be labeled as positive. Values in the range $[0,1]$ are also possible and we examine the best choice of thresholds through the use of an ROC curve that measures the ratio of true positives to false positives as the threshold varies from zero to one.

We used kNN as a baseline for comparison. We identify the *k* nearest neighbors using the distance metric specified in Equation 3. Because the true labels for the individual instances are unknown, multiple instances in a bag are all assumed to have the same label as the bag. If the instances are not individually enumerated, we assign the label to the graph representing the bag itself and use this larger graph for the kNN calculations. We modify the prediction mechanism of kNN in the following manner. For each instance in an unlabeled bag, we determine the ratio of positive instances in the *k* nearest instances. The most extreme of these ratios weighted by the number of different positive or

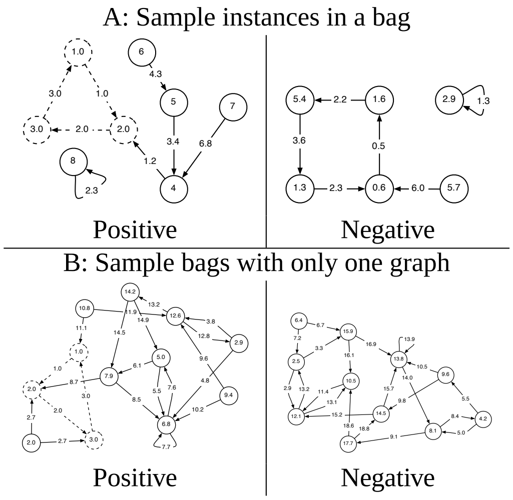


Figure 3. A: Example instances for the three-clique task for the representation where each instance is enumerated. The target concept is shown with dashes. B: Example bags for the three-clique task for the representation where each bag consists of a single large graph.

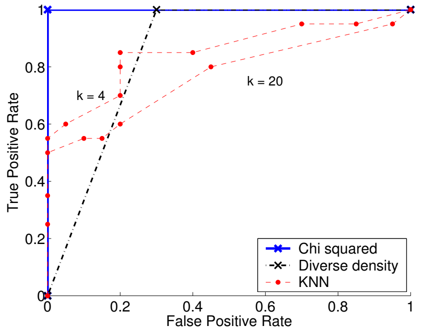


Figure 4. ROC curve comparing the performance of chi-squared, diverse density, and kNN on the illustrative data set. In this case, each bag had an enumerated set of instances.

negative bags that contributed to the ratio is chosen and the ratio itself (without the weighting) is output as the label. The idea of weighting the ratio this way is related to diverse density and helps to make kNN a higher performing baseline for comparison.

Figure 4 shows the ROC curves for relational diverse density, chi-squared, and two values of k for kNN for the first data representation, where there are multiple enumerated instances per bag. These numbers are averaged over 10-folds of cross validation. The test set for each fold was 2 positive bags and 2 negative bags and the training set was the remaining 18 positive bags and 18 negative bags. The chi-squared method identifies the correct target concept each time and had perfect prediction for this task. We do not claim that the chi-squared method will always have perfect performance but it was able to quickly find the target concept for this task. The relational diverse density approach sometimes found a subset of the true concept which gave it a small false positive rate depending on the threshold chosen to differentiate between positive and negative predictions. The two kNN approaches shown had higher accuracy than diverse density for very high thresholds but quickly degraded in performance while diverse density was more robust to threshold changes. At a threshold of 0.5, the accuracies were: chi-squared = 100%, diverse density = 85%, and kNN = 80% and 70% for $k = 4$ and $k = 20$. Accuracy is the percent of bag labels in the test set that are predicted correctly.

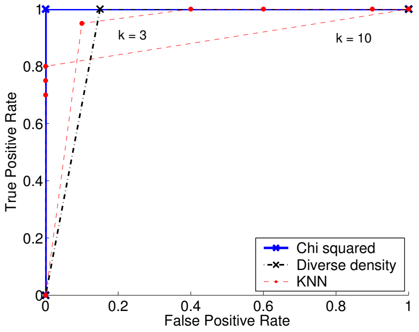
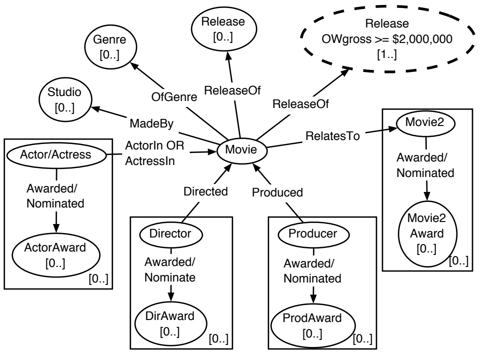


Figure 5. ROC curves for the illustrative data set where each bag had one large graph.

Figure 5 shows the ROC curve for the same three methods in the case where each bag had only one large graph in it. With a threshold of 0.5, chi-squared had 100% accuracy, diverse density had 92.5% accuracy, and kNN had 70% and 50% accuracies for $k = 3$ and $k = 10$. These results are comparable with those presented above and demonstrate that both data representations can be used successfully for MIL on relational data. Our next experiments focus on a much larger database.

5. Experimental results: IMDb

The IMDb is a much larger database with one million objects and nearly five million links. This is a large database where the ability to identify predictive structures should help us to better understand the nature of the database. The two tasks that we present are: predicting which movies will be nominated for academy awards and predicting which movies will gross at least two million dollars (adjusted for inflation) during opening weekend. Both of these tasks are very difficult and if there were a perfect predictor of movie success, then studio executives would have identified it already. Also, both tasks rely on an unknown number of factors which may not all be in the database (e.g., Hollywood politics are not included in IMDb). However, the difficulty



Query constraints:
movie2.year, release.year, Actor-award.year $<$ movie.year
Director-award.year, Producer-award.year $<$ movie.year
$1970 \leq$ movie.year ≤ 2000

Figure 6. qGraph query used to identify high-grossing movies and to create the positive bags. Dashed circles indicate the query restriction and number ranges indicate the minimum number of objects required for a match.

of the tasks provides a good challenge for our approach.

5.1. High-grossing movies

The IMDb is a large connected database and thus corresponds to the second data representation where each bag contains only one instance. We created the bags by querying the database using the qGraph query shown in Figure 6. This query is the depth-two structure surrounding high-grossing movies with the exception that we do not follow links from studios. Studios typically make hundreds of movies and following those links would lead to unnecessarily large graphs. This query returns a set of subgraphs from the database that match the specified structure. In particular, each subgraph will contain a central movie object and its related release objects where at least one release grossed more than 2 million dollars on opening weekend. In addition, any associated studios, genres, producers, directors, actors/actresses, and related movies will be included in each subgraph. If any of the producers, directors, actors/actresses, or related movies have award objects linked to them, these will also be included. Finally, the graph is pruned to remove any events that occurred after the movie's release. This is necessary because we want the structures that the MI learner identifies to predict forward in time. To help minimize noise and the size of the data, we further restrict the set to only contain movies from 1970 to 2000. We randomly sampled this set to obtain approximately 200 positive instances. We reused the same query structure to generate the negative bags except that the releases on opening weekend were restricted to gross less than 2 million dollars. There are a considerable number of such movies so we randomly subsampled to obtain approximately 200 negative bags.

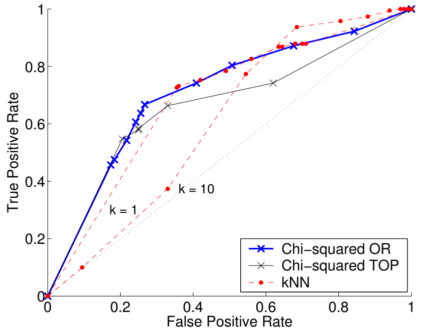


Figure 7. ROC curves comparing the false positive and true positive ratios for the chi-squared MI technique and kNN on the task of predicting high-grossing movies.

We again ran 10-fold cross validation and obtained predictions for the unseen positive and negative bags from the top five percent of the concepts identified by MIL where the concepts are ranked by their chi-squared values. The inability to prune with diverse density hinders its use on such a large data set so we used only the chi-squared approach.

Figure 7 shows the results of this experiment using ROC curves. The chi-square method was able to detect several substructures that predicted high-grossing movies. The results shown in this graph are for the most highly ranked concept on each of the 10 folds, labeled chi-squared TOP, and for the top 5% of the concepts, labeled chi-squared OR. In the latter case, each concept outputs a separate prediction and we used the OR, or max, of these predictions. Although MIL has slightly lower performance in the region of the ROC curve with higher true positives but also higher false positives, its performance is better than kNN in the region with lower false positives and higher true positives. Also, its performance only degrades as the threshold is dropped almost to zero while kNN is less robust to the threshold value. With a threshold of 0.5, chi-squared TOP achieves an accuracy of 69.2% and chi-squared OR has a 70.1% accuracy. kNN's accuracies are 61% and 53.6% for $k = 1$ and $k = 10$. With this prediction mechanism, studios could better allocate money to movies. As we said in the beginning of this experiment, predicting high-grossing movies is a difficult task and it is unlikely that any learning agent could achieve high accuracy.

One of the other benefits of using MIL on this database, besides prediction accuracy, is that the answers are in the form of understandable structures. Figure 8 shows some of the top structures for predicting high-grossing movies. It seems that movies are more likely to be high-grossing if they are related to two or three other movies (e.g. a movie

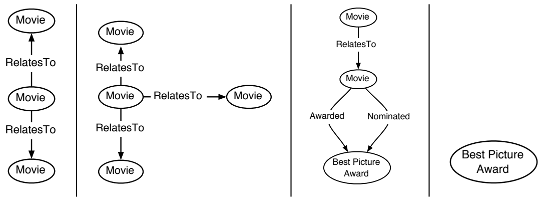


Figure 8. Top predictive relational structures identified by the MI learner on the high-grossing movie task.

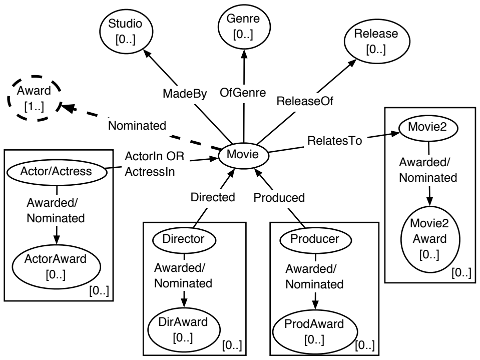
in a series like *Star Trek* or *Indiana Jones* or movies that remade previous successful movies). Another predictive structure that the chi-squared MI learner identified was a movie related to another movie that was both nominated and awarded an academy award for best picture. Last, just the presence of a best picture award object in the subgraph was predictive of movie success.

5.2. Movies nominated for academy awards

We repeated the same experiments for the difficult task of predicting which movies will be nominated for academy awards each year. The query used to generate the positive bags is shown in Figure 9. The structure of this query is identical to that discussed for high-grossing movies except that we require an academy award nomination. The positive bags do not actually contain the award objects for the central movie because we want MIL to identify predictive structures. This query yields 72 positive bags. We use the same query minus the requirement for the awards to create the negative bags. The number of movies which are not nominated for academy awards is quite large and we randomly sample this set to obtain approximately the same number of negative bags (74).

We again compare the predictive ability of chi-squared to kNN on 10-fold cross validation with this data set. These results are shown in Figure 10, again using ROC curves. In this case, the structures found by the MI learner dominate any of the predictions from kNN for all values of k (We show two of the best values of k in the figure). Assuming a threshold of 0.5, the accuracy of chi-squared TOP is 93% and the accuracy of chi-squared OR is 77%. kNN has an accuracy of 49.7% and 50.7% for $k = 5$ and $k = 10$.

We also examine the relational structures that the MI learner identified as predictive of whether a movie will be nominated for an academy award. Some of the top structures are shown in Figure 11. For this task, it seems that movies with at least 20 actors in them are more likely to be nominated for academy awards. This is surprising and is likely due to a reverse effect that better movies have more information in IMDb which means they tend to have more actors associated with them. A related structure has the same form but restricts the genre to drama. These structures may not help a studio executive to better allocate money to new movies but they did identify an important characteristic of the database, which is one of the goals of this work. The presence of only a drama object is enough to predict the nomination in many cases. Last, if a previous award object existed in the subgraph, e.g., if the movie was related to a movie that was also nominated or won an academy award, it was likely to be nominated itself.



Query constraints:
movie2.year, release-year, Actor-award.year $<$ movie.year
Director-award.year, Producer-award.year $<$ movie.year
1970 $\leq$ movie.year $\leq$ 2000

Figure 9. qGraph query used to create the positive bags for the task of predicting which movies will be nominated for an academy award.

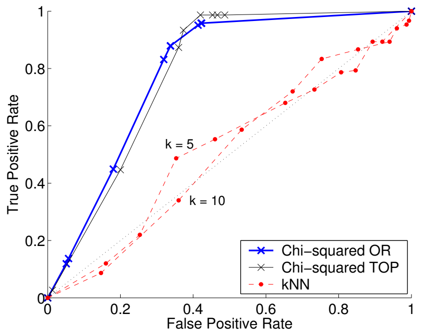


Figure 10. ROC curves for the task of predicting which movies will be nominated for academy awards.

6. Discussion and Conclusions

In this paper, we have presented an approach to identifying predictive structures in relational databases based on the MIL framework. We adapted this framework for use with relational data in two related ways: one where the bags had multiple independent graphs as the instances and one where the bags had one larger graph and the instances were the (implicit) subgraphs of this graph. We demon-

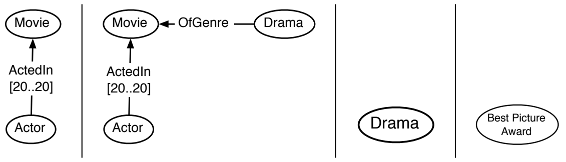


Figure 11. Relational structures identified by the MI learner for predicting academy award nominees.

stated that these adaptations could be used to modify existing MI methods and that the relational version of these methods could be used successfully on both a small and a large database.

One of the strengths of MIL that is emphasized for flat data is the ability to identify which features of the task are important. In the diverse density framework, this is referred to as *scaling*. When the concept is a feature vector, diverse density can identify a scale for each feature that maximizes the diverse density value. If a feature is irrelevant, its best scale will be zero. This strength also applies to the techniques that we presented in this paper. Instead of scaling features in a vector, the concepts identified by the relational MI learner will only contain a subset of the objects and links from the bags. This subset represents the more relevant features with respect to the current task.

Another advantage of MI techniques is that they identify an actual concept (or set of concepts) that can be understood by a human. kNN can be used to label new data but it cannot identify aspects of the data that can help a human to better understand the database. With such structures, a human can iteratively refine their understanding of the database and of the tasks at hand.

Relational probability trees (RPT) (Neville et al., 2003) are a related approach in that they have also been developed to identify predictive structure in large relational databases. However, MIL and RPTs express different relational concepts. RPTs are designed to identify structure in a tree form using attributes on objects or links or structure such as the number of outgoing links from an object. Although this can work very well on tasks such as predicting high-grossing movies, RPTs cannot represent graph concepts such as the 3-clique presented in Section 4.

Acknowledgments

The authors are grateful for comments and helpful discussions from Kiri Wagstaff, Jennifer Neville, Hannah Blau, and the anonymous reviewers. This effort is supported by DARPA and AFRL under contracts F30602-00-2-0597 and F30602-01-2-0566. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of DARPA, AFRL, or the U.S. Government.

References

Amar, R. A., Dooly, D. R., Goldman, S. A., & Zhang, Q. (2001). Multiple-instance learning of real-valued data. *Proc. of the 18th International Conference on Machine Learning* (pp. 3–10). Morgan Kaufmann, SF, CA.

Blau, H., Immerman, N., & Jensen, D. (2002). *A visual language for querying and updating graphs* (Technical Report 2002-037). University of Massachusetts Amherst.

Bunke, H., & Shearer, K. (1998). A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, *19*, 255–259.

Dietterich, T. G., Lathrop, R. H., & Lozano-Perez, T. (1997). Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, *89*, 31–71.

Goldman, S., Zhang, Q., Yu, W., & Fritts, J. E. (2002). Content-based image retrieval using multiple-instance learning. *Proc. of the 19th International Conference on Machine Learning* (pp. 682–689). Morgan Kaufmann, SF, CA.

Maron, O. (1998). *Learning from ambiguity*. Doctoral dissertation, Massachusetts Institute of Technology.

Maron, O., & Lozano-Pérez, T. (1998). A framework for multiple-instance learning. *Advances in Neural Information Processing Systems 10* (pp. 570–576). Cambridge, Massachusetts: MIT Press.

Maron, O., & Ratan, A. L. (1998). Multiple-instance learning for natural scene classification. *Proc. of the 15th International Conference on Machine Learning* (pp. 341–349). Morgan Kaufmann, San Francisco, CA.

McGovern, A., & Jensen, D. (2003). *Chi-squared: A simpler evaluation function for multiple-instance learning* (Technical Report 03-14). Computer Science Department, University of Massachusetts Amherst.

Neville, J., Jensen, D., Friedland, L., & Hay, M. (2003). Learning relational probability trees. To appear in the *Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Mateo, California: Morgan Kaufmann Publishers.

Zhang, Q., & Goldman, S. A. (2002). EM-DD: An improved multiple-instance learning technique. *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press.

Zucker, J.-D., & Chevaleyre, Y. (2000). *Solving multiple-instance and multiple-part learning problems with decision trees and decision rules. application to the mutagenesis problem* (Technical Report 6). University of Paris.